Scalable Node Monitoring

Alexander Drotar
University of Colorado at Boulder

Erin Quinn
Fairleigh Dickinson University

Landon Sutherland
New Mexico State University

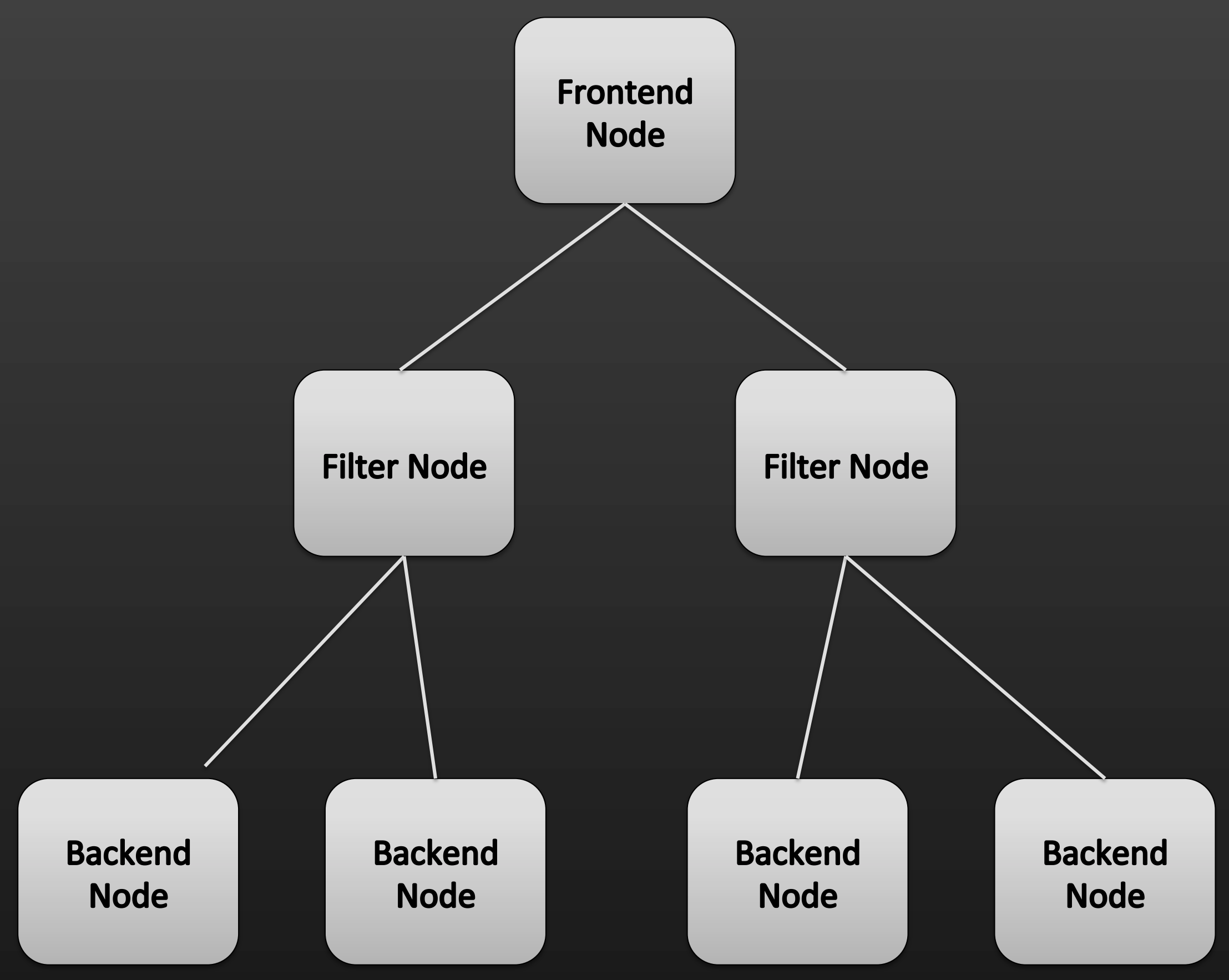
Mentors: Michael Mason - HPC3, Jon Bringhurst - HPC3

Abstract

In the field of cluster computing, node monitoring is crucial. With an abundance of nodes, it is important to make sure that all of the nodes and their applications are running properly to maximize cluster efficiency. Without a node monitoring system in place on a cluster, failed applications can go unnoticed, which can drastically affect performance. This project involved designing and implementing a scalable, parallel tool for monitoring nodes on computer clusters using Component Based Tool Framework (CBTF). This monitoring tool uses Lightweight Data Metric Service (LDMS) to gather node information and using CBTF's inherent scalability, the new tool can operate on a cluster made up of any number of nodes. In this project, the tool was tested and analyzed on an 8-node cluster running Centos 6.2.

Why Change LDMS?

- Fat tree implementation
  - CBTF uses MRNet

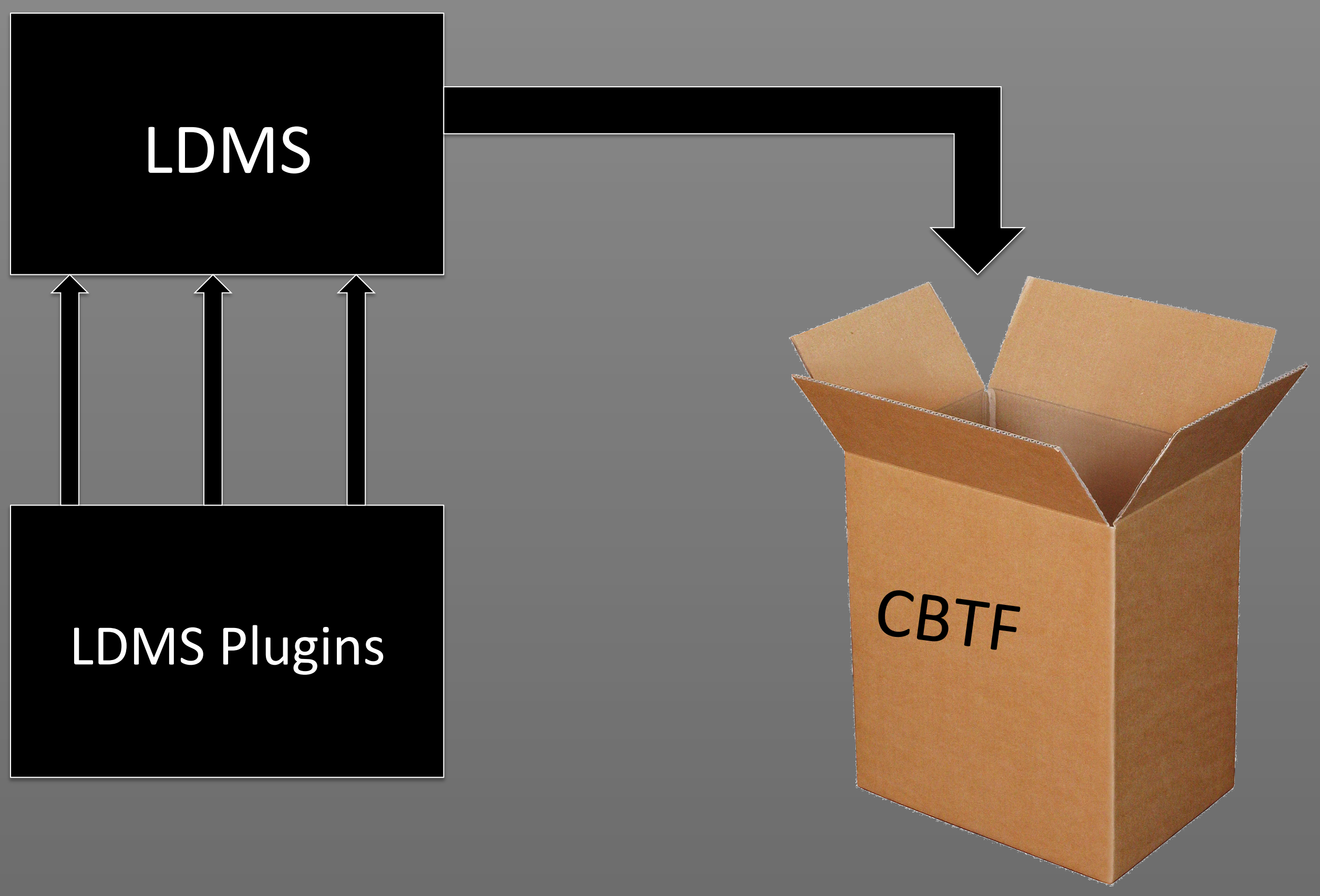


CBTF

- "Component Based Tool Framework"
- Scalable
- Adjusts to different topologies automatically
- Includes a front end node, filter node (s), and backend nodes
- Uses MRNet (Multicast/Reduction Network) mechanism for information transport
- Components can be added very easily to other tools built in CBTF
- Three network levels
  - Frontend Node
  - Filter Nodes
  - Backend Nodes
- Three main files
  - Tool File
  - Component (Plugin) File
  - XML file

LDMS

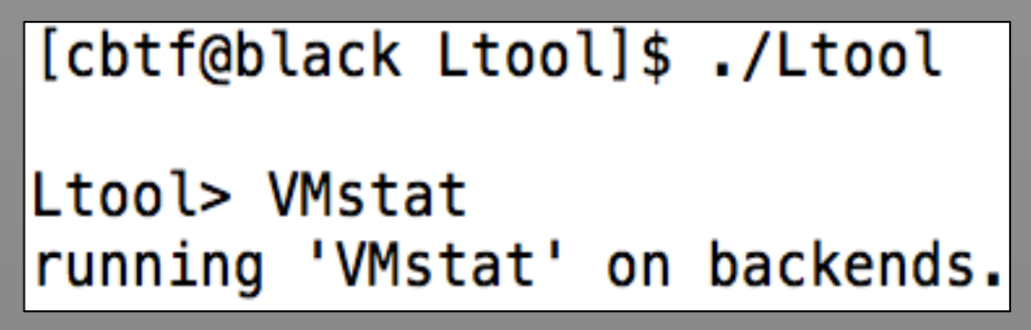
- LDMS stands for "Lightweight Data Metric Service"
- Tool used for monitoring nodes
- Information from /proc/
  - Vmstat, meminfo …



Ltool

Ltool is the name of the tool we derived from LDMS. Its components include the following:

- Vmstat
- Meminfo
- Procinterrupts
- Procstatutil
- Geminfo

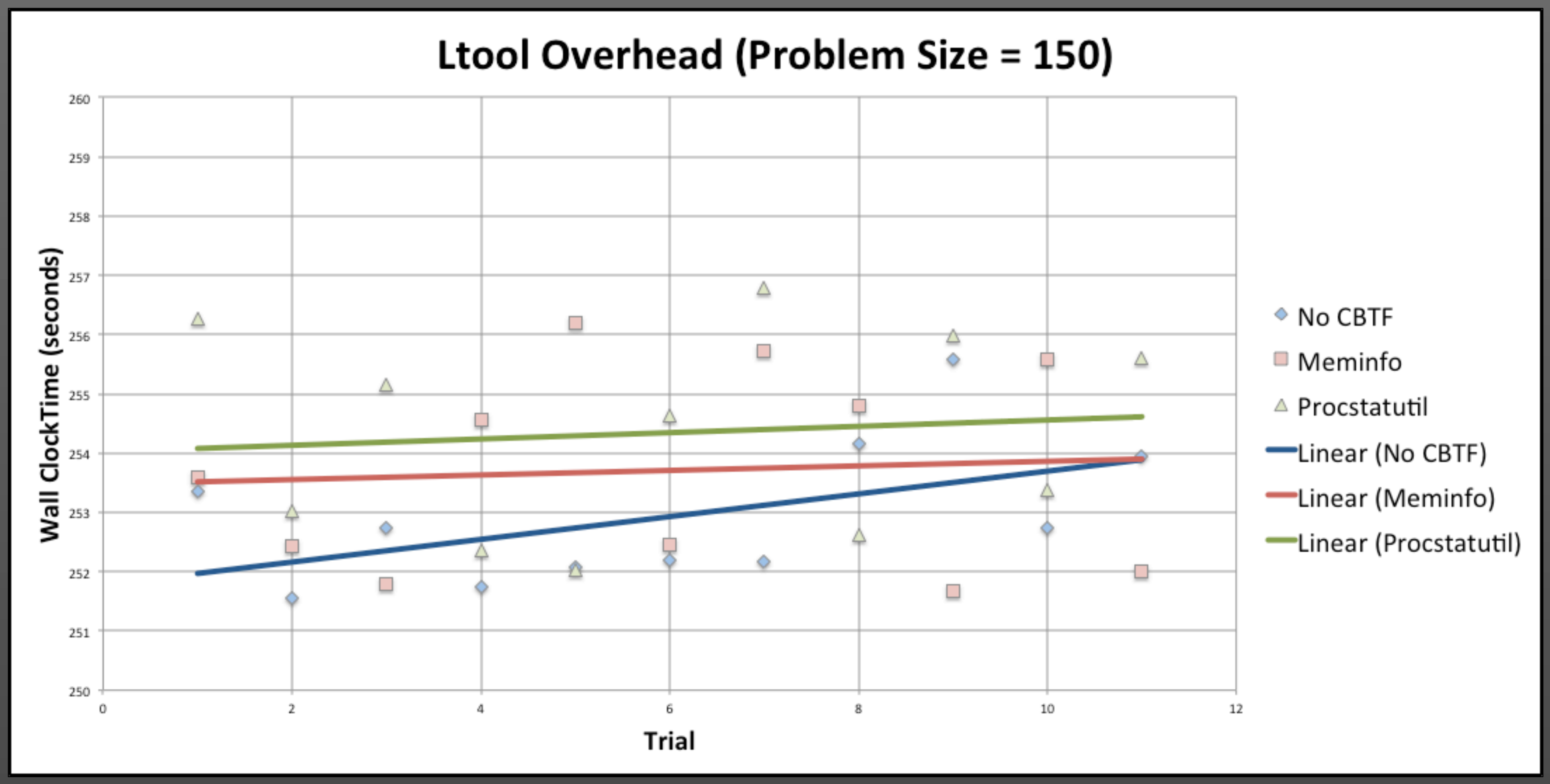


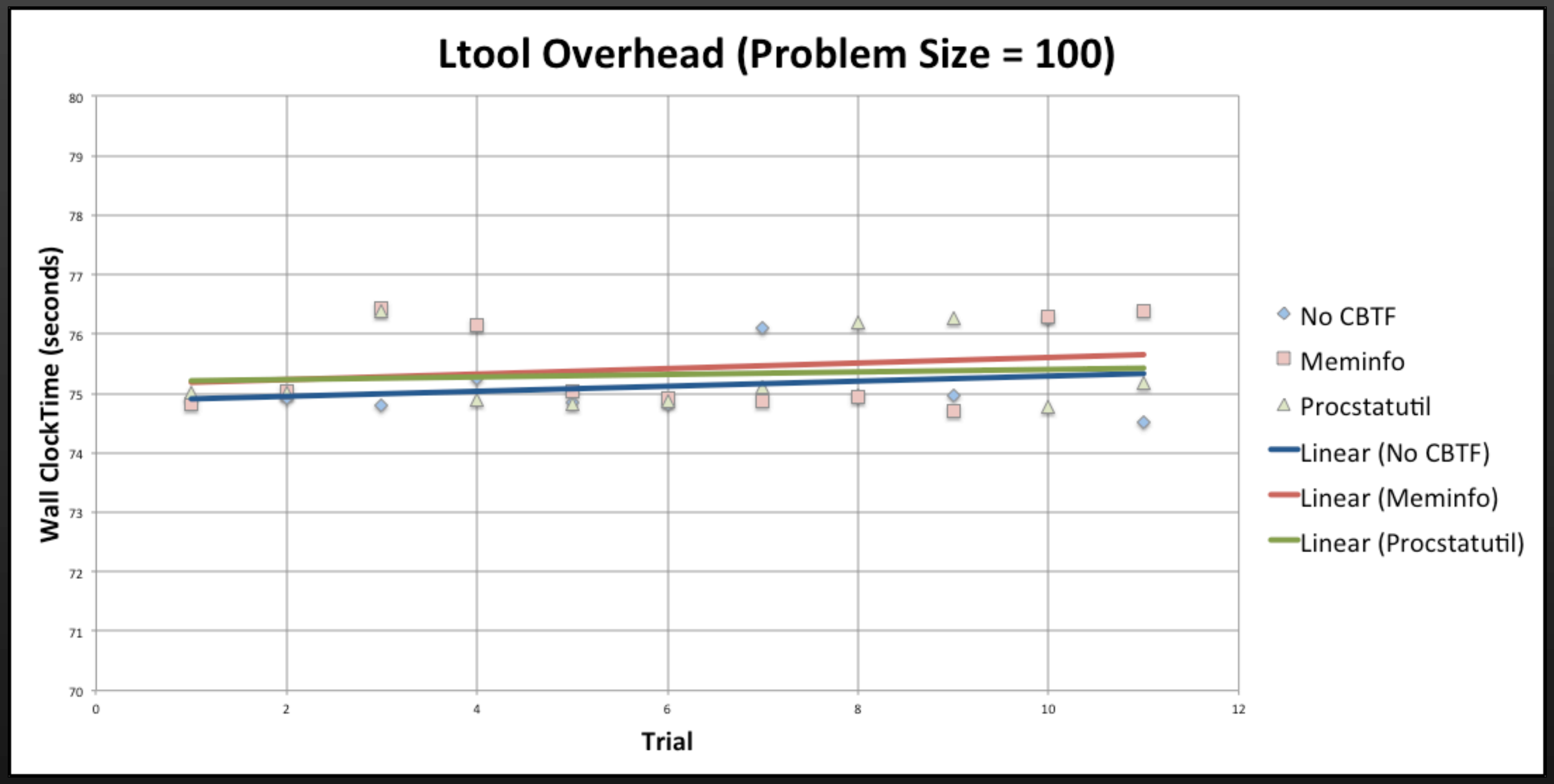
```
[cbtf@black Ltool]$ ./Ltool

Ltool> VMstat
running 'VMstat' on backends.
```

Since CBTF automatically scales, when LDMS is inserted into a CBTF tool, it can be run on a cluster of any size.

Ltool gathers data on backend nodes, concatenates it on filter nodes, and sends the information to a database on the frontend node, where the user can view and query it.





Results

LDMS was successfully implemented into a CBTF tool, and we found that the overhead involved with running the tool was relatively low. This means that running our tool is both efficient and effective, and will not slow down the job on the computer cluster that is running.

Conclusions

Ltool is a useful tool when it comes to monitoring nodes on a cluster because the overhead involved with running the tool is not particularly high and it will automatically scale to any size cluster.

Future Work

- Ltool can be tested to see if it can run consistently for numerous days
- Plugins added to LDMS can be used by Ltool with little modification
- Multiple components executing at the same time
- Use MRNet to filter more data

LA-UR-12-23626

Los Alamos
NATIONAL LABORATORY
EST.1943

New Mexico CONSORTIUM

PRObE
Parallel Reconfigurable Observational Environment

NSF

ISTI
INFORMATION SCIENCE & TECHNOLOGY INSTITUTE